Emulador Cortex-M0

Generado por Doxygen 1.8.10

Domingo, 18 de Octubre de 2015 15:31:05

Índice general

Capítulo 1

Proyecto 1: Emulador Cortex-M0

Emulador Cortex-M0

Materia: Microprocesadores

Codigo destinado a emular el funcionamiento de este procesador.

Autores

John Barahona
Heyler Montoya
Javier Sierra

Fecha

2015 - II

Repositorio disponible en GitHub: `Click Aqui`

`John Alejandro Barahona Pineda - (alejo7053)`

`Heyler Stivens Montoya Orjuela - (heylermontoya)`

`Javier Andres Sierra Pineda - (sierra112)`

Capítulo 2

Índice de estructura de datos

2.1. Estructura de datos

Lista de estructuras con una breve descripción:

Capítulo 3

Indice de archivos

3.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

Capítulo 4

Documentación de las estructuras de datos

4.1. Referencia de la Estructura ins_t

`#include <decoder.h>`

Campos de datos

- char ** array

4.1.1. Documentación de los campos

4.1.1.1. char** array

La documentación para esta estructura fue generada a partir del siguiente fichero:

- decoder.h

4.2. Referencia de la Estructura instruction_t

`#include <decoder.h>`

Campos de datos

- char mnemonic [10]
- char op1_type
- char op2_type
- char op3_type
- uint32_t op1_value
- uint32_t op2_value
- uint32_t op3_value
- uint8_t registers_list [16]

4.2.1. Documentación de los campos

4.2.1.1. char mnemonic[10]

4.2.1.2. char op1_type

4.2.1.3. uint32_t op1_value

4.2.1.4. char op2_type

4.2.1.5. uint32_t op2_value

4.2.1.6. char op3_type

4.2.1.7. uint32_t op3_value

4.2.1.8. uint8_t registers_list[16]

La documentación para esta estructura fue generada a partir del siguiente fichero:

- decoder.h

4.3. Referencia de la Estructura port_t

```
#include <io.h>
```

Campos de datos

- uint8_t DDR
- uint8_t PORT
- uint8_t PIN
- uint8_t Pins
- uint8_t Interrupts

4.3.1. Documentación de los campos

4.3.1.1. uint8_t DDR

4.3.1.2. uint8_t Interrupts

4.3.1.3. uint8_t PIN

4.3.1.4. uint8_t Pins

4.3.1.5. uint8_t PORT

La documentación para esta estructura fue generada a partir del siguiente fichero:

- io.h

Capítulo 5

Documentación de archivos

5.1. Referencia del Archivo decoder.c

```
#include "decoder.h"
```

'defines'

- #define PC 15

  *Macro que define la posicion del registro **PC** en el puntero **dir_reg**.*

Funciones

- void decodeInstruction (instruction_t instruction, uint32_t ∗dir_reg, char ∗dir_flags, uint8_t ∗SRAM, uint16_t ∗dec)
- instruction_t getInstruction (char ∗instStr)

  Obtiene la instrucción separada por partes.
- int readFile (char ∗filename, ins_t ∗instructions)
- int countLines (FILE ∗fp)

Variables

- uint8_t data
- uint16_t ∗ dec

5.1.1. Documentación de los 'defines'

5.1.1.1. #define PC 15

Macro que define la posicion del registro ***PC*** *en* el puntero **dir_reg**.

5.1.2. Documentación de las funciones

5.1.2.1. int countLines (FILE ∗ *fp*)

5.1.2.2. void decodeInstruction (instruction_t *instruction,* uint32_t ∗ *dir_reg,* char ∗ *dir_flags,* uint8_t ∗ *SRAM,* uint16_t ∗ *dec*)

5.1.2.3. instruction_t getInstruction (char * *instStr*)

Obtiene la instrucción separada por partes.

Parámetros

instStr	cadena que contiene la instrucción.

Devuelve

instruction_t la instrucción separada por partes.

5.1.2.4. int readFile (char * *filename,* ins_t * *instructions*)

5.1.3. Documentación de las variables

5.1.3.1. uint8_t data

5.1.3.2. uint16_t* dec

5.2. Referencia del Archivo decoder.h

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include "instrucciones.h"
#include "instruc_desplazamiento.h"
#include "io.h"
#include "salto.h"
#include <curses.h>
```

Estructuras de datos

- struct ins_t
- struct instruction_t

Funciones

- void decodeInstruction (instruction_t instruction, uint32_t *dir_reg, char *dir_flags, uint8_t *SRAM, uint16_t *dec)
- instruction_t getInstruction (char *instStr)

  Obtiene la instrucción separada por partes.
- int readFile (char *filename, ins_t *instructions)
- int countLines (FILE *fp)

5.2.1. Documentación de las funciones

5.2.1.1. int countLines (FILE * *fp*)

5.2.1.2. void decodeInstruction (instruction_t *instruction,* uint32_t * *dir_reg,* char * *dir_flags,* uint8_t * *SRAM,* uint16_t * *dec*)

5.2.1.3. instruction_t getInstruction (char * *instStr*)

Obtiene la instrucción separada por partes.

Parámetros

instStr	cadena que contiene la instrucción.

Devuelve

instruction_t la instrucción separada por partes.

5.2.1.4. int readFile (char * *filename,* ins_t * *instructions*)

5.3. Referencia del Archivo imprimir.c

```
#include "imprimir.h"
```

Funciones

- void valor_registro (uint32_t valores[ ])

valor_registro

prototipo de la funcion valor_registro que muestra el valor que tienen los registros

5.3.1. Documentación de las funciones

5.3.1.1. void valor_registro (uint32_t *valores[ ]*)

valor_registro

prototipo de la funcion valor_registro que muestra el valor que tienen los registros

5.4. Referencia del Archivo imprimir.h

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <curses.h>
```

Funciones

- void valor_registro (uint32_t valores[ ])

valor_registro

prototipo de la funcion valor_registro que muestra el valor que tienen los registros

5.4.1. Documentación de las funciones

5.4.1.1. void valor_registro (uint32_t *valores[ ]*)

valor_registro

prototipo de la funcion valor_registro que muestra el valor que tienen los registros

5.5. Referencia del Archivo instruc_desplazamiento.c

```
#include "instruc_desplazamiento.h"
```

'defines'

- #define N 0

  *Macro que define la posicion de la bandera **N** en el puntero **dir_flags**.*
- #define Z 1

  *Macro que define la posicion de la bandera **Z** en el puntero **dir_flags**.*
- #define C 2

  *Macro que define la posicion de la bandera **C** en el puntero **dir_flags**.*
- #define V 3

  *Macro que define la posicion de la bandera **V** en el puntero **dir_flags**.*

Funciones

- uint32_t LSL (uint32_t Rdn, uint32_t Rm, char ∗dir_flags)

  LSL

  prototipo de la funcion de desplazamiento LSL que va a operar los registros.
- uint32_t LSR (uint32_t Rdn, uint32_t Rm, char ∗dir_flags)

  LSR

  prototipo de la funcion de desplazamiento LSR que va a operar los registros
- uint32_t ROR (uint32_t Rdn, uint32_t Rm, char ∗dir_flags)

  ROR

  prototipo de la funcion de desplazamiento ROR que va a operar los registros
- uint32_t ASR (uint32_t Rdn, uint32_t Rm, char ∗dir_flags)

  ASR

  prototipo de la funcion de desplazamiento ASR que va a operar los registros

- uint32_t BIC (uint32_t Rdn, uint32_t Rm, char ∗dir_flags)

  BIC

  prototipo de la funcion de desplazamiento BIC que va a operar los registros

- uint32_t MVN (uint32_t Rdn, char ∗dir_flags)

  MVN

  prototipo de la funcion de desplazamiento MVN que va a operar los registros

- uint32_t RSB (uint32_t Rdn, char ∗dir_flags)

  RSB

  prototipo de la funcion de desplazamiento RSB que va a operar los registros

- uint32_t REV (uint32_t Rdn)

  REV

  prototipo de la funcion de desplazamiento REV que va a operar los registros

- uint32_t REVG (uint32_t Rdn)

  REVG

  prototipo de la funcion de desplazamiento REV que va a operar los registros

- uint32_t REVSH (uint32_t Rdn)

  REVSH

  prototipo de la funcion de desplazamiento REV que va a operar los registros

5.5.1. Documentación de los 'defines'

5.5.1.1. #define C 2

Macro que define la posicion de la bandera ***C*** *en* el puntero **dir_flags**.

5.5.1.2. #define N 0

Macro que define la posicion de la bandera ***N*** *en* el puntero **dir_flags**.

5.5.1.3. #define V 3

Macro que define la posicion de la bandera ***V*** *en* el puntero **dir_flags**.

5.5.1.4. #define Z 1

Macro que define la posicion de la bandera ***Z*** *en* el puntero **dir_flags**.

5.5.2. Documentación de las funciones

5.5.2.1. uint32_t ASR (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

ASR

prototipo de la funcion de desplazamiento ASR que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion ASR

5.5.2.2. uint32_t BIC (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

BIC

prototipo de la funcion de desplazamiento BIC que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion BIC

5.5.2.3. uint32_t LSL (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

LSL

prototipo de la funcion de desplazamiento LSL que va a operar los registros.

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion LSL

5.5.2.4. uint32_t LSR (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

LSR

prototipo de la funcion de desplazamiento LSR que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion LSR

5.5.2.5. uint32_t MVN (uint32_t *Rdn,* char * *dir_flags*)

MVN

prototipo de la funcion de desplazamiento MVN que va a operar los registros

Parámetros

Rdn	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion MVN

5.5.2.6. uint32_t REV (uint32_t *Rdn*)

REV

prototipo de la funcion de desplazamiento REV que va a operar los registros

Parámetros

Rdn	operando ingresado

Devuelve

retorna el resultado de la operacion REV

5.5.2.7. uint32_t REVG (uint32_t *Rdn*)

REVG

prototipo de la funcion de desplazamiento REV que va a operar los registros

Parámetros

Rdn	operando ingresado

Devuelve

retorna el resultado de la operacion REVG

5.5.2.8. uint32_t REVSH (uint32_t *Rdn*)

REVSH

prototipo de la funcion de desplazamiento REV que va a operar los registros

Parámetros

Rdn	operando ingresado

Devuelve

retorna el resultado de la operacion REVSH

5.5.2.9. uint32_t ROR (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

ROR

prototipo de la funcion de desplazamiento ROR que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion ROR

5.5.2.10. uint32_t RSB (uint32_t *Rdn,* char * *dir_flags*)

RSB

prototipo de la funcion de desplazamiento RSB que va a operar los registros

Parámetros

Rdn	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion RSB

5.6. Referencia del Archivo instruc_desplazamiento.h

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include "instrucciones.h"
```

Funciones

- uint32_t LSL (uint32_t Rdn, uint32_t Rm, char *dir_flags)

LSL

prototipo de la funcion de desplazamiento LSL que va a operar los registros.

- uint32_t LSR (uint32_t Rdn, uint32_t Rm, char *dir_flags)

LSR

prototipo de la funcion de desplazamiento LSR que va a operar los registros

- uint32_t ROR (uint32_t Rdn, uint32_t Rm, char *dir_flags)

ROR

prototipo de la funcion de desplazamiento ROR que va a operar los registros

- uint32_t ASR (uint32_t Rdn, uint32_t Rm, char *dir_flags)

ASR

prototipo de la funcion de desplazamiento ASR que va a operar los registros

- uint32_t BIC (uint32_t Rdn, uint32_t Rm, char *dir_flags)

BIC

prototipo de la funcion de desplazamiento BIC que va a operar los registros

- uint32_t MVN (uint32_t Rdn, char *dir_flags)

MVN

prototipo de la funcion de desplazamiento MVN que va a operar los registros

- uint32_t RSB (uint32_t Rdn, char *dir_flags)

RSB

prototipo de la funcion de desplazamiento RSB que va a operar los registros

- uint32_t REV (uint32_t Rdn)

REV

prototipo de la funcion de desplazamiento REV que va a operar los registros

- uint32_t REVG (uint32_t Rdn)

REVG

prototipo de la funcion de desplazamiento REV que va a operar los registros

- uint32_t REVSH (uint32_t Rdn)

REVSH

prototipo de la funcion de desplazamiento REV que va a operar los registros

5.6.1. Documentación de las funciones

5.6.1.1. uint32_t ASR (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

ASR

prototipo de la funcion de desplazamiento ASR que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion ASR

5.6.1.2. uint32_t BIC (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

BIC

prototipo de la funcion de desplazamiento BIC que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion BIC

5.6.1.3. uint32_t LSL (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

LSL

prototipo de la funcion de desplazamiento LSL que va a operar los registros.

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion LSL

5.6.1.4. uint32_t LSR (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

LSR

prototipo de la funcion de desplazamiento LSR que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion LSR

5.6.1.5. uint32_t MVN (uint32_t *Rdn,* char * *dir_flags*)

MVN

prototipo de la funcion de desplazamiento MVN que va a operar los registros

Parámetros

Rdn	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion MVN

5.6.1.6. uint32_t REV (uint32_t *Rdn*)

REV

prototipo de la funcion de desplazamiento REV que va a operar los registros

Parámetros

Rdn	operando ingresado

Devuelve

retorna el resultado de la operacion REV

5.6.1.7. uint32_t REVG (uint32_t *Rdn*)

REVG

prototipo de la funcion de desplazamiento REV que va a operar los registros

Parámetros

Rdn	operando ingresado

Devuelve

retorna el resultado de la operacion REVG

5.6.1.8. uint32_t REVSH (uint32_t *Rdn*)

REVSH

prototipo de la funcion de desplazamiento REV que va a operar los registros

Parámetros

Rdn	operando ingresado

Devuelve

retorna el resultado de la operacion REVSH

5.6.1.9. uint32_t ROR (uint32_t *Rdn,* uint32_t *Rm,* char * *dir_flags*)

ROR

prototipo de la funcion de desplazamiento ROR que va a operar los registros

Parámetros

Rdn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion ROR

5.6.1.10. uint32_t RSB (uint32_t *Rdn,* char * *dir_flags*)

RSB

prototipo de la funcion de desplazamiento RSB que va a operar los registros

Parámetros

Rdn	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion RSB

5.7. Referencia del Archivo instrucciones.c

```
#include "instrucciones.h"
```

'defines'

- #define N 0

  *Macro que define la posicion de la bandera **N** en el puntero **dir_flags**.*
- #define Z 1

*Macro que define la posicion de la bandera **Z** en el puntero **dir_flags**.*

- #define C 2

  *Macro que define la posicion de la bandera **C** en el puntero **dir_flags**.*
- #define V 3

  *Macro que define la posicion de la bandera **V** en el puntero **dir_flags**.*
- #define PC 15

  *Macro que define la posicion del registro **PC** en el puntero **dir_reg**.*
- #define LR 14

  *Macro que define la posicion del registro **LR** en el puntero **dir_reg**.*
- #define SP 13

  *Macro que define la posicion del registro **SP** en el puntero **dir_reg**.*

Funciones

- void flags (uint32_t Rn, uint32_t Rm, uint32_t Rd, char ∗dir_flags)

  flags

  prototipo de la funcion flags que va a operar los registros de las banderas

- void flag_N (uint32_t Rd, char ∗dir_flag_N)

  flag_N

  prototipo de la funcion flag_N que va a operar el registro de la bandera N

- void flag_Z (uint32_t Rd, char ∗dir_flag_Z)

  flag_Z

  prototipo de la funcion flag_Z que va a operar el registro de la bandera Z

- void flag_C (uint32_t Rn, uint32_t Rm, uint32_t Rd, char ∗dir_flag_C)

  flag_C

  prototipo de la funcion flag_C que va a operar el registro de la bandera C

- void flag_V (uint32_t Rn, uint32_t Rm, uint32_t Rd, char ∗dir_flag_V)

  flag_V

  prototipo de la funcion flag_V que va a operar el registro de la bandera V

- void CMN (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

  CMN

  prototipo de la funcion CMN que va a operar los registros

- void CMP (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

  CMP

  prototipo de la funcion CMP que va a operar los registros

- uint32_t MUL (uint32_t Rn, uint32_t Rm, char *dir_flags)

  MUL

  prototipo de la funcion MUL que va a operar los registros

- void TST (uint32_t Rn, uint32_t Rm, char *dir_flags)

  TST

  prototipo de la funcion TST que va a operar los registros

- uint32_t ADD (uint32_t Rn, uint32_t Rm, char *dir_flags)

  ADD

  prototipo de la funcion ADD que va a operar los registros

- uint32_t ADC (uint32_t Rn, uint32_t Rm, char *dir_flags)

  ADC

  prototipo de la funcion ADC que va a operar los registros

- uint32_t SUB (uint32_t Rn, uint32_t Rm, char *dir_flags)

  SUB

  prototipo de la funcion SUB que va a operar los registros

- uint32_t SBC (uint32_t Rn, uint32_t Rm, char *dir_flags)

  SBC

  prototipo de la funcion SBC que va a operar los registros

- uint32_t MOV (uint32_t Rn, char *dir_flags)

  MOV

  prototipo de la funcion MOV que va a operar los registros

- uint32_t AND (uint32_t Rn, uint32_t Rm, char *dir_flags)

  AND

  prototipo de la funcion AND que va a operar los registros

- uint32_t EOR (uint32_t Rn, uint32_t Rm, char *dir_flags)

  EOR

  prototipo de la funcion EOR que va a operar los registros

- uint32_t ORR (uint32_t Rn, uint32_t Rm, char *dir_flags)

  ORR

  prototipo de la funcion ORR que va a operar los registros

- void NOP (uint32_t *dir_reg)

  NOP

  prototipo de la funcion NOP que va a operar los registros
- void PUSH (uint8_t *SRAM, uint32_t *dir_reg, uint8_t *R_activos)

  Función PUSH

  Guarda datos en la Pila
- uint32_t bitcount (uint8_t *R)

  bitcount

  funcion bitcount que va a operar los registros contando cuantos estan en 1
- void POP (uint8_t *SRAM, uint32_t *dir_reg, uint8_t *R_activos)

  POP

  Extrae los registros de la pila
- uint32_t LDR (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 32bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.
- uint32_t LDRB (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.
- uint32_t LDRH (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 16 bits de SRAM,los guarda en un registro de 32 bits haciendo extension de cero.
- uint32_t LDRSB (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.
- uint32_t LDRSH (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 16 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.
- void STR (uint32_t Rt, uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Guarda 32 bits de un registro en SRAM.
- void STRB (uint32_t Rt, uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Guarda los primeros 8 bits de un registro en SRAM.
- void STRH (uint32_t Rt, uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Guarda los primeros 16 bits de un registro en SRAM.
- void PUSHI (uint8_t *SRAM, uint32_t *dir_reg, char *dir_flags)

  Guarda datos especificos en la pila que son dados por las interrupciones.
- void POPI (uint8_t *SRAM, uint32_t *dir_reg, char *dir_flags)

  Extrae datos especificos despues de la interrupcion.
- void NVIC (uint8_t *IRQ, uint8_t *SRAM, uint32_t *dir_reg, char *dir_flags)

  Interrupciones.

5.7.1. Documentación de los 'defines'

5.7.1.1. #define C 2

Macro que define la posicion de la bandera ***C*** *en* el puntero **dir_flags**.

5.7.1.2. #define LR 14

Macro que define la posicion del registro ***LR*** *en* el puntero **dir_reg**.

5.7.1.3. #define N 0

Macro que define la posicion de la bandera ***N*** *en* el puntero **dir_flags**.

5.7.1.4. #define PC 15

Macro que define la posicion del registro ***PC*** *en* el puntero **dir_reg**.

5.7.1.5. #define SP 13

Macro que define la posicion del registro ***SP*** *en* el puntero **dir_reg**.

5.7.1.6. #define V 3

Macro que define la posicion de la bandera ***V*** *en* el puntero **dir_flags**.

5.7.1.7. #define Z 1

Macro que define la posicion de la bandera ***Z*** *en* el puntero **dir_flags**.

5.7.2. Documentación de las funciones

5.7.2.1. uint32_t ADC (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

ADC

prototipo de la funcion ADC que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresaaado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la ADC

5.7.2.2. uint32_t ADD (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

ADD

prototipo de la funcion ADD que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresaaado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la ADD

5.7.2.3. uint32_t AND (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

AND

prototipo de la funcion AND que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion AND

5.7.2.4. uint32_t bitcount (uint8_t * *R*)

bitcount

funcion bitcount que va a operar los registros contando cuantos estan en 1

Parámetros

R	registros que va a operar la funcion

Devuelve

retorna el numero de registro activos

5.7.2.5. void CMN (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

CMN

prototipo de la funcion CMN que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.7.2.6. void CMP (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

CMP

prototipo de la funcion CMP que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.7.2.7. uint32_t EOR (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

EOR

prototipo de la funcion EOR que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion EOR

5.7.2.8. void flag_C (uint32_t *Rn,* uint32_t *Rm,* uint32_t *Rd,* char * *dir_flag_C*)

flag_C

prototipo de la funcion flag_C que va a operar el registro de la bandera C

Parámetros

Rn	operando ingresado
Rm	operando ingresado
Rd	operando ingresado
dir_flag_C	puntero que tiene la direccion de las banderas

5.7.2.9. void flag_N (uint32_t *Rd,* char * *dir_flag_N*)

flag_N

prototipo de la funcion flag_N que va a operar el registro de la bandera N

Parámetros

Rd	operando ingresado
dir_flag_N	puntero que tiene la direccion de las banderas

5.7.2.10. void flag_V (uint32_t *Rn,* uint32_t *Rm,* uint32_t *Rd,* char * *dir_flag_V*)

flag_V

prototipo de la funcion flag_V que va a operar el registro de la bandera V

Parámetros

Rn	operando ingresado
Rm	operando ingresado
Rd	operando ingresado
dir_flag_V	puntero que tiene la direccion de las banderas

5.7.2.11. void flag_Z (uint32_t *Rd,* char * *dir_flag_Z*)

flag_Z

prototipo de la funcion flag_Z que va a operar el registro de la bandera Z

Parámetros

Rd	operando ingresado
dir_flag_Z	puntero que tiene la direccion de las banderas

5.7.2.12. void flags (uint32_t *Rn,* uint32_t *Rm,* uint32_t *Rd,* char * *dir_flags*)

flags

prototipo de la funcion flags que va a operar los registros de las banderas

Parámetros

Rn	operando ingresado
Rm	operando ingresado
Rd	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.7.2.13. uint32_t LDR (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 32bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.7.2.14. uint32_t LDRB (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.7.2.15. uint32_t LDRH (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 16 bits de SRAM,los guarda en un registro de 32 bits haciendo extension de cero.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.7.2.16. uint32_t LDRSB (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.7.2.17. uint32_t LDRSH (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 16 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.7.2.18. uint32_t MOV (uint32_t *Rn,* char * *dir_flags*)

MOV

prototipo de la funcion MOV que va a operar los registros

Parámetros

Rn	operando ingresado

dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion MOV

5.7.2.19. uint32_t MUL (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

MUL

prototipo de la funcion MUL que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion de las MUL

5.7.2.20. void NOP (uint32_t * *dir_reg*)

NOP

prototipo de la funcion NOP que va a operar los registros

Parámetros

dir_reg	puntero que tiene la direccion de los registros

5.7.2.21. void NVIC (uint8_t * *IRQ,* uint8_t * *SRAM,* uint32_t * *dir_reg,* char * *dir_flags*)

Interrupciones.

Parámetros

IRQ	puntero al arreglo de interrupciones

SRAM	puntero a la memoria SRAM
dir_reg	puntero a los registro principales
dir_flags	puntero a las banderas

5.7.2.22. uint32_t ORR (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

ORR

prototipo de la funcion ORR que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion ORR

5.7.2.23. void POP (uint8_t * *SRAM,* uint32_t * *dir_reg,* uint8_t * *R_activos*)

POP

Extrae los registros de la pila

Parámetros

SRAM	registros de SRAM que va a operar la funcion
dir_reg	puntero a registros
R_activos	puntero a registros activos

5.7.2.24. void POPI (uint8_t * *SRAM,* uint32_t * *dir_reg,* char * *dir_flags*)

Extrae datos especificos despues de la interrupcion.

Parámetros

SRAM	puntero a la memoria SRAM
dir_reg	puntero a los registro principales
dir_flags	puntero a las banderas

5.7.2.25. void PUSH (uint8_t * *SRAM,* uint32_t * *dir_reg,* uint8_t * *R_activos*)

Función PUSH

Guarda datos en la Pila

Parámetros

SRAM	puntero a los registros de esta memoria
dir_reg	puntero a los registro principales
R_activos	puntero a los registros activos

5.7.2.26. void PUSHI (uint8_t * *SRAM,* uint32_t * *dir_reg,* char * *dir_flags*)

Guarda datos especificos en la pila que son dados por las interrupciones.

Parámetros

SRAM	puntero a la memoria SRAM
dir_reg	puntero a los registro principales
dir_flags	puntero a las banderas

5.7.2.27. uint32_t SBC (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

SBC

prototipo de la funcion SBC que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresaaado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la SBC

5.7.2.28. void STR (uint32_t *Rt,* uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Guarda 32 bits de un registro en SRAM.

Parámetros

Rt	registro a guardar en SRAM
Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

5.7.2.29. void STRB (uint32_t *Rt,* uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Guarda los primeros 8 bits de un registro en SRAM.

Parámetros

Rt	registro a guardar en SRAM
Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

5.7.2.30. void STRH (uint32_t *Rt,* uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Guarda los primeros 16 bits de un registro en SRAM.

Parámetros

Rt	registro a guardar en SRAM
Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

5.7.2.31. uint32_t SUB (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

SUB

prototipo de la funcion SUB que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la SUB

5.7.2.32. void TST (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

TST

prototipo de la funcion TST que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.8. Referencia del Archivo instrucciones.h

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
```

Funciones

- void flags (uint32_t Rn, uint32_t Rm, uint32_t Rd, char ∗dir_flags)

  flags

  prototipo de la funcion flags que va a operar los registros de las banderas

- void flag_N (uint32_t Rd, char ∗dir_flag_N)

  flag_N

  prototipo de la funcion flag_N que va a operar el registro de la bandera N

- void flag_Z (uint32_t Rd, char ∗dir_flag_Z)

  flag_Z

  prototipo de la funcion flag_Z que va a operar el registro de la bandera Z

- void flag_C (uint32_t Rn, uint32_t Rm, uint32_t Rd, char ∗dir_flag_C)

  flag_C

  prototipo de la funcion flag_C que va a operar el registro de la bandera C

- void flag_V (uint32_t Rn, uint32_t Rm, uint32_t Rd, char ∗dir_flag_V)

  flag_V

  prototipo de la funcion flag_V que va a operar el registro de la bandera V

- void CMN (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

  CMN

  prototipo de la funcion CMN que va a operar los registros

- void CMP (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

CMP

prototipo de la funcion CMP que va a operar los registros

- uint32_t MUL (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

MUL

prototipo de la funcion MUL que va a operar los registros

- void TST (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

TST

prototipo de la funcion TST que va a operar los registros

- uint32_t ADD (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

ADD

prototipo de la funcion ADD que va a operar los registros

- uint32_t ADC (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

ADC

prototipo de la funcion ADC que va a operar los registros

- uint32_t SUB (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

SUB

prototipo de la funcion SUB que va a operar los registros

- uint32_t SBC (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

SBC

prototipo de la funcion SBC que va a operar los registros

- uint32_t MOV (uint32_t Rn, char ∗dir_flags)

MOV

prototipo de la funcion MOV que va a operar los registros

- uint32_t AND (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

AND

prototipo de la funcion AND que va a operar los registros

- uint32_t EOR (uint32_t Rn, uint32_t Rm, char ∗dir_flags)

EOR

prototipo de la funcion EOR que va a operar los registros

- uint32_t ORR (uint32_t Rn, uint32_t Rm, char *dir_flags)

  ORR

  prototipo de la funcion ORR que va a operar los registros
- void NOP (uint32_t *dir_reg)

  NOP

  prototipo de la funcion NOP que va a operar los registros
- void PUSH (uint8_t *SRAM, uint32_t *dir_reg, uint8_t *R_activos)

  Función PUSH

  Guarda datos en la Pila
- uint32_t bitcount (uint8_t *R)

  bitcount

  funcion bitcount que va a operar los registros contando cuantos estan en 1
- void POP (uint8_t *SRAM, uint32_t *dir_reg, uint8_t *R_activos)

  POP

  Extrae los registros de la pila
- uint32_t LDR (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 32bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.
- uint32_t LDRB (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.
- uint32_t LDRH (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 16 bits de SRAM,los guarda en un registro de 32 bits haciendo extension de cero.
- uint32_t LDRSB (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.
- uint32_t LDRSH (uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Extrae 16 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.
- void STR (uint32_t Rt, uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Guarda 32 bits de un registro en SRAM.
- void STRB (uint32_t Rt, uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Guarda los primeros 8 bits de un registro en SRAM.
- void STRH (uint32_t Rt, uint32_t Rn, uint32_t Rm, uint8_t *SRAM)

  Guarda los primeros 16 bits de un registro en SRAM.
- void NVIC (uint8_t *IRQ, uint8_t *SRAM, uint32_t *dir_reg, char *dir_flags)

  Interrupciones.
- void PUSHI (uint8_t *SRAM, uint32_t *dir_reg, char *dir_flags)

  Guarda datos especificos en la pila que son dados por las interrupciones.
- void POPI (uint8_t *SRAM, uint32_t *dir_reg, char *dir_flags)

  Extrae datos especificos despues de la interrupcion.

5.8.1. Documentación de las funciones

5.8.1.1. uint32_t ADC (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

ADC

prototipo de la funcion ADC que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresaaado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la ADC

5.8.1.2. uint32_t ADD (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

ADD

prototipo de la funcion ADD que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresaaado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la ADD

5.8.1.3. uint32_t AND (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

AND

prototipo de la funcion AND que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion AND

5.8.1.4. uint32_t bitcount (uint8_t * R)

bitcount

funcion bitcount que va a operar los registros contando cuantos estan en 1

Parámetros

R	registros que va a operar la funcion

Devuelve

retorna el numero de registro activos

5.8.1.5. void CMN (uint32_t Rn, uint32_t Rm, char * dir_flags)

CMN

prototipo de la funcion CMN que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.8.1.6. void CMP (uint32_t Rn, uint32_t Rm, char * dir_flags)

CMP

prototipo de la funcion CMP que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.8.1.7. uint32_t EOR (uint32_t Rn, uint32_t Rm, char * dir_flags)

EOR

prototipo de la funcion EOR que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion EOR

5.8.1.8. void flag_C (uint32_t *Rn,* uint32_t *Rm,* uint32_t *Rd,* char * *dir_flag_C*)

flag_C

prototipo de la funcion flag_C que va a operar el registro de la bandera C

Parámetros

Rn	operando ingresado
Rm	operando ingresado
Rd	operando ingresado
dir_flag_C	puntero que tiene la direccion de las banderas

5.8.1.9. void flag_N (uint32_t *Rd,* char * *dir_flag_N*)

flag_N

prototipo de la funcion flag_N que va a operar el registro de la bandera N

Parámetros

Rd	operando ingresado
dir_flag_N	puntero que tiene la direccion de las banderas

5.8.1.10. void flag_V (uint32_t *Rn,* uint32_t *Rm,* uint32_t *Rd,* char * *dir_flag_V*)

flag_V

prototipo de la funcion flag_V que va a operar el registro de la bandera V

Parámetros

Rn	operando ingresado
Rm	operando ingresado
Rd	operando ingresado
dir_flag_V	puntero que tiene la direccion de las banderas

5.8.1.11. void flag_Z (uint32_t *Rd,* char * *dir_flag_Z*)

flag_Z

prototipo de la funcion flag_Z que va a operar el registro de la bandera Z

Parámetros

Rd	operando ingresado
dir_flag_Z	puntero que tiene la direccion de las banderas

5.8.1.12. void flags (uint32_t *Rn,* uint32_t *Rm,* uint32_t *Rd,* char * *dir_flags*)

flags

prototipo de la funcion flags que va a operar los registros de las banderas

Parámetros

Rn	operando ingresado
Rm	operando ingresado
Rd	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.8.1.13. uint32_t LDR (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 32bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.8.1.14. uint32_t LDRB (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de cero.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.8.1.15. uint32_t LDRH (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 16 bits de SRAM,los guarda en un registro de 32 bits haciendo extension de cero.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.8.1.16. uint32_t LDRSB (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 8 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.8.1.17. uint32_t LDRSH (uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Extrae 16 bits de SRAM, los guarda en un registro de 32 bits haciendo extension de signo.

Parámetros

Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

Devuelve

Rt registro donde guarda los datos extraidos

5.8.1.18. uint32_t MOV (uint32_t *Rn*, char * *dir_flags*)

MOV

prototipo de la funcion MOV que va a operar los registros

Parámetros

Rn	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion MOV

5.8.1.19. uint32_t MUL (uint32_t *Rn*, uint32_t *Rm*, char * *dir_flags*)

MUL

prototipo de la funcion MUL que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion de las MUL

5.8.1.20. void NOP (uint32_t * *dir_reg*)

NOP

prototipo de la funcion NOP que va a operar los registros

Parámetros

dir_reg	puntero que tiene la direccion de los registros

5.8.1.21. void NVIC (uint8_t * *IRQ*, uint8_t * *SRAM*, uint32_t * *dir_reg*, char * *dir_flags*)

Interrupciones.

Parámetros

IRQ	puntero al arreglo de interrupciones
SRAM	puntero a la memoria SRAM
dir_reg	puntero a los registro principales
dir_flags	puntero a las banderas

5.8.1.22. uint32_t ORR (uint32_t *Rn*, uint32_t *Rm*, char * *dir_flags*)

ORR

prototipo de la funcion ORR que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la operacion ORR

5.8.1.23. void POP (uint8_t * *SRAM*, uint32_t * *dir_reg*, uint8_t * *R_activos*)

POP

Extrae los registros de la pila

Parámetros

SRAM	registros de SRAM que va a operar la funcion
dir_reg	puntero a registros
R_activos	puntero a registros activos

5.8.1.24. void POPI (uint8_t * *SRAM,* uint32_t * *dir_reg,* char * *dir_flags*)

Extrae datos especificos despues de la interrupcion.

Parámetros

SRAM	puntero a la memoria SRAM
dir_reg	puntero a los registro principales
dir_flags	puntero a las banderas

5.8.1.25. void PUSH (uint8_t * *SRAM,* uint32_t * *dir_reg,* uint8_t * *R_activos*)

Función PUSH

Guarda datos en la Pila

Parámetros

SRAM	puntero a los registros de esta memoria
dir_reg	puntero a los registro principales
R_activos	puntero a los registros activos

5.8.1.26. void PUSHI (uint8_t * *SRAM,* uint32_t * *dir_reg,* char * *dir_flags*)

Guarda datos especificos en la pila que son dados por las interrupciones.

Parámetros

SRAM	puntero a la memoria SRAM
dir_reg	puntero a los registro principales
dir_flags	puntero a las banderas

5.8.1.27. uint32_t SBC (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

SBC

prototipo de la funcion SBC que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresaaado

dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la SBC

5.8.1.28. void STR (uint32_t *Rt,* uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Guarda 32 bits de un registro en SRAM.

Parámetros

Rt	registro a guardar en SRAM
Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

5.8.1.29. void STRB (uint32_t *Rt,* uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Guarda los primeros 8 bits de un registro en SRAM.

Parámetros

Rt	registro a guardar en SRAM
Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

5.8.1.30. void STRH (uint32_t *Rt,* uint32_t *Rn,* uint32_t *Rm,* uint8_t * *SRAM*)

Guarda los primeros 16 bits de un registro en SRAM.

Parámetros

Rt	registro a guardar en SRAM
Rn	direccion de memoria
Rm	numero o direccion de memoria
SRAM	puntero a la memoria SRAM

5.8.1.31. uint32_t SUB (uint32_t *Rn,* uint32_t *Rm,* char * *dir_flags*)

SUB

prototipo de la funcion SUB que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

Devuelve

retorna el resultado de la SUB

5.8.1.32. void TST (uint32_t *Rn*, uint32_t *Rm*, char * *dir_flags*)

TST

prototipo de la funcion TST que va a operar los registros

Parámetros

Rn	operando ingresado
Rm	operando ingresado
dir_flags	puntero que tiene la direccion de las banderas

5.9. Referencia del Archivo io.c

```
#include "io.h"
```

Funciones

- void initIO (void)

  Inicializa los puertos de Entrada y Salida.
- void changePinPortA (uint8_t pin, uint8_t value)

  Cambia el valor de los puertos A.
- void changePinPortB (uint8_t pin, uint8_t value)

  Cambia el valor de los puertos B.
- void IOAccess (uint8_t address, uint8_t *data, uint8_t r_w)

  Escribe datos en el DDRB.
- void showPorts (void)

  Muestra los puertos en la interfaz.
- void showFrame (int x, int y, int w, int h)

  Organiza la interfaz de los puertos.

Variables

- port_t PORTA
- port_t PORTB

5.9.1. Documentación de las funciones

5.9.1.1. void changePinPortA (uint8_t *pin*, uint8_t *value*)

Cambia el valor de los puertos A.

Parámetros

pin	numero del pin
value	valor del pin

5.9.1.2. void changePinPortB (uint8_t *pin*, uint8_t *value*)

Cambia el valor de los puertos B.

Parámetros

pin	numero del pin
value	valor del pin

5.9.1.3. void initIO (void)

Inicializa los puertos de Entrada y Salida.

5.9.1.4. void IOAccess (uint8_t *address*, uint8_t * *data*, uint8_t *r_w*)

Escribe datos en el DDRB.

Parámetros

address	direccion de memoria
data	puntero al dato
r_w	lee o escribe

5.9.1.5. void showFrame (int *x*, int *y*, int *w*, int *h*)

Organiza la interfaz de los puertos.

Parámetros

x	coordenada X
y	coordenada Y
w	
h	

5.9.1.6. void showPorts (void)

Muestra los puertos en la interfaz.

5.9.2. Documentación de las variables

5.9.2.1. port_t PORTA

5.9.2.2. port_t PORTB

5.10. Referencia del Archivo io.h

```
#include <stdint.h>
#include <curses.h>
```

Estructuras de datos

- struct port_t

'defines'

- #define XINIT 10
- #define YINIT 5
- #define HIGH 1
- #define LOW 0
- #define Read 1
- #define Write 0
- #define BLUEBLACK 10 /∗Text Blue Background Black∗/
- #define REDBLACK 20 /∗Text Red Background Black∗/
- #define WHITEBLACK 30 /∗Text White Background White∗/

Funciones

- void IOAccess (uint8_t address, uint8_t ∗data, uint8_t r_w)

  Escribe datos en el DDRB.
- void changePinPortA (uint8_t pin, uint8_t value)

  Cambia el valor de los puertos A.
- void changePinPortB (uint8_t pin, uint8_t value)

  Cambia el valor de los puertos B.
- void initIO (void)

  Inicializa los puertos de Entrada y Salida.
- void showPorts (void)

  Muestra los puertos en la interfaz.
- void showFrame (int x, int y, int w, int h)

  Organiza la interfaz de los puertos.

Variables

- uint8_t IRQ [16]

5.10.1. Documentación de los 'defines'

5.10.1.1. #define BLUEBLACK 10 /∗Text Blue Background Black∗/

5.10.1.2. #define HIGH 1

5.10.1.3. #define LOW 0

5.10.1.4. #define Read 1

5.10.1.5. #define REDBLACK 20 /∗Text Red Background Black∗/

5.10.1.6. #define WHITEBLACK 30 /∗Text White Background White∗/

5.10.1.7. #define Write 0

5.10.1.8. #define XINIT 10

5.10.1.9. #define YINIT 5

5.10.2. Documentación de las funciones

5.10.2.1. void changePinPortA (uint8_t *pin,* uint8_t *value*)

Cambia el valor de los puertos A.

Parámetros

pin	numero del pin
value	valor del pin

5.10.2.2. void changePinPortB (uint8_t *pin,* uint8_t *value*)

Cambia el valor de los puertos B.

Parámetros

pin	numero del pin
value	valor del pin

5.10.2.3. void initIO (void)

Inicializa los puertos de Entrada y Salida.

5.10.2.4. void IOAccess (uint8_t *address,* uint8_t ∗ *data,* uint8_t *r_w*)

Escribe datos en el DDRB.

Parámetros

address	direccion de memoria
data	puntero al dato

r_w	lee o escribe

5.10.2.5. void showFrame (int *x*, int *y*, int *w*, int *h*)

Organiza la interfaz de los puertos.

Parámetros

x	coordenada X
y	coordenada Y
w	
h	

5.10.2.6. void showPorts (void)

Muestra los puertos en la interfaz.

5.10.3. Documentación de las variables

5.10.3.1. uint8_t IRQ[16]

5.11. Referencia del Archivo main.c

```
#include <curses.h>
#include "imprimir.h"
#include "instrucciones.h"
#include "instruc_desplazamiento.h"
#include "salto.h"
#include "decoder.h"
#include "io.h"
#include <time.h>
```

'defines'

- #define N 0

  *Macro que define la posicion de la bandera **N** en el registro**APSR**.*
- #define Z 1

  *Macro que define la posicion de la bandera **Z** en el registro**APSR**.*
- #define C 2

  *Macro que define la posicion de la bandera **C** en el registro **APSR**.*
- #define V 3

  *Macro que define la posicion de la bandera **V** en el registro **APSR**.*
- #define LR 14

  *Macro que define la posicion del registro **LR** en el arreglo **R**.*
- #define PC 15

  *Macro que define la posicion del registro **PC** en el arreglo **R**.*
- #define SP 13

  *Macro que define la posicion del registro **SP** en el arreglo **R**.*
- #define TAM_SRAM 255

  *Macro que define el tamaño del arreglo que emula la memoria **SRAM**.*

Funciones

- int main (void)

5.11.1. Documentación de los 'defines'

5.11.1.1. #define C 2

Macro que define la posicion de la bandera ***C*** *en* el registro ***APSR***.

5.11.1.2. #define LR 14

Macro que define la posicion del registro ***LR*** *en* el arreglo ***R***.

5.11.1.3. #define N 0

Macro que define la posicion de la bandera ***N*** *en* el registro***APSR***.

5.11.1.4. #define PC 15

Macro que define la posicion del registro ***PC*** *en* el arreglo ***R***.

5.11.1.5. #define SP 13

Macro que define la posicion del registro ***SP*** *en* el arreglo ***R***.

5.11.1.6. #define TAM_SRAM 255

Macro que define el tamaño del arreglo que emula la memoria ***SRAM***.

5.11.1.7. #define V 3

Macro que define la posicion de la bandera ***V*** *en* el registro ***APSR***.

5.11.1.8. #define Z 1

Macro que define la posicion de la bandera ***Z*** *en* el registro***APSR***.

5.11.2. Documentación de las funciones

5.11.2.1. int main (void)

5.12. Referencia del Archivo README.md

5.13. Referencia del Archivo salto.c

```
#include "salto.h"
```

'defines'

- #define N 0

  *Macro que define la posicion de la bandera **N** en el puntero **dir_flags**.*
- #define Z 1

  *Macro que define la posicion de la bandera **Z** en el puntero **dir_flags**.*
- #define C 2

  *Macro que define la posicion de la bandera **C** en el puntero **dir_flags**.*
- #define V 3

  *Macro que define la posicion de la bandera **V** en el puntero **dir_flags**.*
- #define LR 14

  *Macro que define la posicion del registro **LR** en el puntero **dir_reg**.*
- #define PC 15

  *Macro que define la posicion del registro **PC** en el puntero **dir_reg**.*

Funciones

- void B (uint32_t label, uint32_t ∗dir_reg)

  B

  prototipo de la funcion B que va a modificar el registro PC
- void BL (uint32_t label, uint32_t ∗dir_reg)

  BL

  prototipo de la funcion BL que va a modificar el registro PC
- void BX (uint32_t ∗dir_reg)

  BX

  prototipo de la funcion BX que va a modificar el registro PC
- void BEQ (uint32_t label, uint32_t ∗dir_reg, char ∗dir_flags)

  BEQ

  prototipo de la funcion BEQ que va a modificar el registro PC
- void BNE (uint32_t label, uint32_t ∗dir_reg, char ∗dir_flags)

  BNE

  prototipo de la funcion BNE que va a modificar el registro PC
- void BCS (uint32_t label, uint32_t ∗dir_reg, char ∗dir_flags)

  BCS

  prototipo de la funcion BCS que va a modificar el registro PC
- void BCC (uint32_t label, uint32_t ∗dir_reg, char ∗dir_flags)

BCC

prototipo de la funcion BCC que va a modificar el registro PC

- void BMI (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BMI

prototipo de la funcion BMI que va a modificar el registro PC

- void BPL (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BPL

prototipo de la funcion BPL que va a modificar el registro PC

- void BVS (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BVS

prototipo de la funcion BVS que va a modificar el registro PC

- void BVC (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BVC

prototipo de la funcion BVC que va a modificar el registro PC

- void BHI (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BHI

prototipo de la funcion BHI que va a modificar el registro PC

- void BLS (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BLS

prototipo de la funcion BLS que va a modificar el registro PC

- void BGE (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BGE

prototipo de la funcion BGE que va a modificar el registro PC

- void BLT (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BLT

prototipo de la funcion BLT que va a modificar el registro PC

- void BGT (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BGT

prototipo de la funcion BGT que va a modificar el registro PC

- void BLE (uint32_t label, uint32_t ∗dir_reg, char ∗dir_flags)

BLE

prototipo de la funcion BLE que va a modificar el registro PC

- void BAL (uint32_t label, uint32_t ∗dir_reg)

BAL

prototipo de la funcion BAL que va a modificar el registro PC

5.13.1. Documentación de los 'defines'

5.13.1.1. #define C 2

Macro que define la posicion de la bandera ***C*** *en* el puntero **dir_flags**.

5.13.1.2. #define LR 14

Macro que define la posicion del registro ***LR*** *en* el puntero **dir_reg**.

5.13.1.3. #define N 0

Macro que define la posicion de la bandera ***N*** *en* el puntero **dir_flags**.

5.13.1.4. #define PC 15

Macro que define la posicion del registro ***PC*** *en* el puntero **dir_reg**.

5.13.1.5. #define V 3

Macro que define la posicion de la bandera ***V*** *en* el puntero **dir_flags**.

5.13.1.6. #define Z 1

Macro que define la posicion de la bandera ***Z*** *en* el puntero **dir_flags**.

5.13.2. Documentación de las funciones

5.13.2.1. void B (uint32_t *label,* uint32_t ∗ *dir_reg*)

B

prototipo de la funcion B que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros

5.13.2.2. void BAL (uint32_t *label*, uint32_t * *dir_reg*)

BAL

prototipo de la funcion BAL que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros

5.13.2.3. void BCC (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BCC

prototipo de la funcion BCC que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.4. void BCS (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BCS

prototipo de la funcion BCS que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.5. void BEQ (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BEQ

prototipo de la funcion BEQ que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.6. void BGE (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BGE

prototipo de la funcion BGE que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.7. void BGT (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BGT

prototipo de la funcion BGT que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.8. void BHI (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BHI

prototipo de la funcion BHI que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.9. void BL (uint32_t *label,* uint32_t * *dir_reg*)

BL

prototipo de la funcion BL que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros

5.13.2.10. void BLE (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BLE

prototipo de la funcion BLE que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.11. void BLS (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BLS

prototipo de la funcion BLS que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.12. void BLT (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BLT

prototipo de la funcion BLT que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.13. void BMI (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BMI

prototipo de la funcion BMI que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.14. void BNE (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BNE

prototipo de la funcion BNE que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.15. void BPL (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BPL

prototipo de la funcion BPL que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.16. void BVC (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BVC

prototipo de la funcion BVC que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.17. void BVS (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BVS

prototipo de la funcion BVS que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.13.2.18. void BX (uint32_t * *dir_reg*)

BX

prototipo de la funcion BX que va a modificar el registro PC

Parámetros

dir_reg	puntero que tiene la direccion de los registros

5.14. Referencia del Archivo salto.h

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
```

Funciones

- void B (uint32_t label, uint32_t *dir_reg)

B

prototipo de la funcion B que va a modificar el registro PC

- void BL (uint32_t label, uint32_t *dir_reg)

BL

prototipo de la funcion BL que va a modificar el registro PC

- void BX (uint32_t *dir_reg)

BX

prototipo de la funcion BX que va a modificar el registro PC

- void BEQ (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BEQ

prototipo de la funcion BEQ que va a modificar el registro PC

- void BNE (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BNE

prototipo de la funcion BNE que va a modificar el registro PC

- void BCS (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BCS

prototipo de la funcion BCS que va a modificar el registro PC

- void BCC (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BCC

prototipo de la funcion BCC que va a modificar el registro PC

- void BMI (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BMI

prototipo de la funcion BMI que va a modificar el registro PC

- void BPL (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BPL

prototipo de la funcion BPL que va a modificar el registro PC

- void BVS (uint32_t label, uint32_t *dir_reg, char *dir_flags)

BVS

prototipo de la funcion BVS que va a modificar el registro PC

- void BVC (uint32_t label, uint32_t *dir_reg, char *dir_flags)

  BVC

  prototipo de la funcion BVC que va a modificar el registro PC

- void BHI (uint32_t label, uint32_t *dir_reg, char *dir_flags)

  BHI

  prototipo de la funcion BHI que va a modificar el registro PC

- void BLS (uint32_t label, uint32_t *dir_reg, char *dir_flags)

  BLS

  prototipo de la funcion BLS que va a modificar el registro PC

- void BGE (uint32_t label, uint32_t *dir_reg, char *dir_flags)

  BGE

  prototipo de la funcion BGE que va a modificar el registro PC

- void BLT (uint32_t label, uint32_t *dir_reg, char *dir_flags)

  BLT

  prototipo de la funcion BLT que va a modificar el registro PC

- void BGT (uint32_t label, uint32_t *dir_reg, char *dir_flags)

  BGT

  prototipo de la funcion BGT que va a modificar el registro PC

- void BLE (uint32_t label, uint32_t *dir_reg, char *dir_flags)

  BLE

  prototipo de la funcion BLE que va a modificar el registro PC

- void BAL (uint32_t label, uint32_t *dir_reg)

  BAL

  prototipo de la funcion BAL que va a modificar el registro PC

5.14.1. Documentación de las funciones

5.14.1.1. void B (uint32_t *label,* uint32_t * *dir_reg*)

B

prototipo de la funcion B que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros

5.14.1.2. void BAL (uint32_t *label*, uint32_t * *dir_reg*)

BAL

prototipo de la funcion BAL que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros

5.14.1.3. void BCC (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BCC

prototipo de la funcion BCC que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.4. void BCS (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BCS

prototipo de la funcion BCS que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.5. void BEQ (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BEQ

prototipo de la funcion BEQ que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.6. void BGE (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BGE

prototipo de la funcion BGE que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.7. void BGT (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BGT

prototipo de la funcion BGT que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.8. void BHI (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BHI

prototipo de la funcion BHI que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.9. void BL (uint32_t *label,* uint32_t * *dir_reg*)

BL

prototipo de la funcion BL que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros

5.14.1.10. void BLE (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BLE

prototipo de la funcion BLE que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.11. void BLS (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BLS

prototipo de la funcion BLS que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.12. void BLT (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BLT

prototipo de la funcion BLT que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.13. void BMI (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BMI

prototipo de la funcion BMI que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.14. void BNE (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BNE

prototipo de la funcion BNE que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.15. void BPL (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BPL

prototipo de la funcion BPL que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.16. void BVC (uint32_t *label*, uint32_t * *dir_reg*, char * *dir_flags*)

BVC

prototipo de la funcion BVC que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.17. void BVS (uint32_t *label,* uint32_t * *dir_reg,* char * *dir_flags*)

BVS

prototipo de la funcion BVS que va a modificar el registro PC

Parámetros

label	operando ingresado
dir_reg	puntero que tiene la direccion de los registros
dir_flags	puntero que tiene la direccion de las banderas

5.14.1.18. void BX (uint32_t * *dir_reg*)

BX

prototipo de la funcion BX que va a modificar el registro PC

Parámetros

dir_reg	puntero que tiene la direccion de los registros

5.15. Referencia del Archivo test.c

```
#include "imprimir.h"
#include "instrucciones.h"
#include "instruc_desplazamiento.h"
#include <curses.h>
```

'defines'

- #define N 0

  *Macro que define la posicion de la bandera **N** en el registro**regf**.*

- #define Z 1

  *Macro que define la posicion de la bandera **Z** en el registro**regf**.*
- #define C 2

  *Macro que define la posicion de la bandera **C** en el registro **regf**.*
- #define V 3

  *Macro que define la posicion de la bandera **V** en el registro **regf**.*

Funciones

- int main ()

  TEXT

  Este archivo es para probar las librerias.

5.15.1. Documentación de los 'defines'

5.15.1.1. #define C 2

Macro que define la posicion de la bandera ***C*** *en* el registro ***regf***.

5.15.1.2. #define N 0

Macro que define la posicion de la bandera ***N*** *en* el registro***regf***.

5.15.1.3. #define V 3

Macro que define la posicion de la bandera ***V*** *en* el registro ***regf***.

5.15.1.4. #define Z 1

Macro que define la posicion de la bandera ***Z*** *en* el registro***regf***.

5.15.2. Documentación de las funciones

5.15.2.1. int main (void)

TEXT

Este archivo es para probar las librerias.